**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION**

**4.1.0 INTRODUCTION**

This chapter focuses on the implementation of the project. The choice of development tools is stated and explained here. How the system is tested, and the various ways it is tested is explained in this chapter. Also, the target computer system requirements are outlined here to give an overview of the requirement necessary to run the program. Finally, the various maintenance process is discussed.

**4.2.0 FEATURES AND CHOICE OF IMPLEMENTATION LANGUAGE**

Python programming language is the language of choice for carrying out this project. It was selected for many reasons which includes the following:

1. It is free
2. Great support for machine learning
3. Runs on a wide variety of system architecture (cross platform)
4. Syntactically simple to understand.

In carrying out this project, there is a need for a programming environment that aids and enable writing python code and displaying its output on the fly. Jupyter notebook proved to be more convenient than any other, because of the important features it has especially its method of displaying output.

The process of determining the presence of heart disease is a form of adaptive programming, where the program extracts information from the data provided and optimize the accuracy of its output in the discovery of knowledge from the data. The data must be structured for the accurate extraction of knowledge from the data.

In this project, the dataset used is structured in a tabular form, by the Panda’s Library. From the website pandas.pydata.org (2017), “Pandas” is an open source BSD Licensed library providing high performance, easy-to-use data structures and data analysis tools for the Python programming language.

There are many machine-learning frameworks used today, one of the very best is the python machine learning framework “Scikit learn”. It has a wide range of basic and advanced machine learning algorithms built in. It has great support (or integration) with the Pandas library. The three algorithms used in this project are from the Scikit learn. From the website scikit-learn.org/stable (2017), Scikit-learn contains simple and efficient tools for data mining and data analysis, which is accessible to everybody and reusable in various contexts.

**4.3.1 SYSTEM TESTING STRATEGIES**

Testing a system is the process using it extensively in order to find errors or any area of malfunction, it is a method of preventive maintenance that ensures the system does what it was meant to do and ascertain that it does it well before deployment to production.

In machine-learning, ways of testing a trained model, may be either based on performance or accuracy. Based on the scope of this project, focus is only on accuracy.

For accuracy testing, two methods have been used to test each trained model. They are namely:

* Train test datasets.
* Cross Validation.

**4.3.1.1 TRAIN TEST DATASETS**

In this type of testing, as the name implies, the dataset is divided into two parts: one is the train set, while the other is the test set. The train set is used to train the algorithm and the test set is passed to the trained model to predict. Thereafter, the prediction result is compared to the result of the test set which is known beforehand. The training set is typically larger than the test set, for example, in this project, the ratio of training set data to test set data is 7:3. i.e the train set is 70% while the test set is 30% of the total data collected. The percentage of correct predictions after the comparison is therefore the accuracy of that particular trained model.

**4.3.1.2 CROSS VALIDATION**

This is a more sophisticated approach than using a test and train dataset. It uses the entire transformed dataset to train and test a given algorithm.

It first involves separating the dataset into a number of equally sized groups of instances (called folds). The model is then trained on all fold’s exception one that was left out and the prepared model is tested on that left out fold. The process is repeated so that each fold gets an opportunity at being left out and acting as the test dataset. Finally, the performance measures are averaged across all folds to estimate the capability of the algorithm on the problem. (Brownlee, 2013b)

**4.3.1.3 CORRELATION**

Correlation is the statistical measure that describes the association between random variables. In some cases, like linear regression, having correlated variable can greatly decrease the overall performance of a machine-learning model (Brownlee, 2016). Correlation between two or more variable in a dataset can cause bias in the model. The correlated data will have more influence on the model. Although sometimes, correlation does not mean causality, It is always proper to ensure there is not much correlation between the data variables. In this project, all values have been tested for correlation. This is to ensure the accuracy and efficiency in performance of the machine-learning algorithms used.

**4.4.0 TARGET COMPUTER SYSTEM REQUIREMENT**

The training of machine-learning algorithms is a CPU-bound process, and therefore can only be carried out by a certain range of hardware. The machine-learning program in this project will run on systems that meet the following specifications.

* Ram of 2GB or above
* Intel Quad Core or AMD A4 or better.
* Storage: Hard disk with size greater than 80GB

Software requirements include the following:

* Operating systems: Microsoft Windows 7 or newer, mac OS, Linux distributions with kernel versions later than 3.0
* Jupyter Notebook.
* Python 2.0 or later.
* Enthought Canopy.
* Web Browser (Google Chrome or any other modern web browser).

All program code was written and tested with Jupyter Notebook running on an Arch Linux distribution and windows operating system running on Intel Core i3 processor. Data for this project was gotten from the UCI machine learning repository (Aha, 2016). According to the website, the database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one that has been used by Machine Learning researchers to this date. The goal field, refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4.

**4.5.0 SOFTWARE MAINTENANCE ISSUES**

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance and or other attributes. (Wikipedia, 2017).

In machine-learning, the most necessary type of maintenance practice is, including the test data to the initial training data and retrain the algorithm to get a better trained model. Since more data increases the accuracy of a machine learning model, adding more data to the train set and reevaluating the model will always yield a more intelligent machine-learning model.

With this kind of maintenance, the machine-learning model tends to increase in accuracy with continuous use. Only correct predictions cases are added to the train set to prevent training the model with wrong data as this will drastically decrease the accuracy of the model and also decrease its reliability.

**CHAPTER FIVE**

**SUMMARY AND CONCLUSION**

**5.1 INTRODUCTION**

This chapter is a summary of the entire project. It concludes the project by explaining the functionalities of the program as well as outlining the various use cases of the program in improving the health sector. Furthermore, recommendations for future research and the conclusion is presented in this chapter.

**5.2 SUMMARY**

This project uses machine learning in determining the likelihood of an individual having a heart disease based on a given set of symptoms. Based on 13 (thirteen) featured symptoms, a score of 1 for presence and 0 for absence of heart disease is given for each individual. Three different machine-learning algorithms were trained with data gotten from the UCI machine learning repository (Aha, 2016).

Due to limited availability of data, two different type of testing method was carried out, namely; the train/test method and the cross-validation method.

The train test method divides the data into train set and test set in 7:3 ratio respectively, the train set is used to train each algorithm while the test set is passed to the algorithm and its prediction is compared to the known result to determine its accuracy.

In the cross-validation method, the dataset is divided into five (five was used in this project) different set, four of them is combined to train the machine-learning algorithm while the last one is used to test the model. This method is repeated until each one of the set has been used to train the model as well as test the model, the average of the accuracy of each training and testing instance is regarded as the accuracy of the model.

The program was written with the python programming language on the Jupyter notebook environment for proper documentation. The following are the machine-learning algorithm used in the project; Naïve Bayes, Logistic regression and Support Vector Machine.

**5.3 RECOMMENDATIONS (FUTURE RESEARCH)**

For future research, it is quite imperative that the following, if considered in furthering this research will greatly enlarge the scope and achieve a better result for various use cases.

* Adding more dataset: If more dataset is used to train a machine-learning algorithm, the derived model will definitely have a higher accuracy. Also having dataset from local areas will also increase the scope of the model in determining the likelihood of having a heart disease in that location.
* Use of more machine learning algorithm: There are numerous machine learning algorithms. In this project, only three of them were tested and used. Testing others may yet reveal a higher accuracy than the once used here, and even have a better performance in terms of CPU and memory usage.
* Implementation of use case systems: System that harness the power of machine learning should be developed to increase efficiency in diagnosing heart disease patient in hospitals. Developing mobile applications that uses machine learning can help users track their health status right on their phones without visiting any health center.

**5.4 CONCLUSION**

In conclusion of this project, testing three algorithms namely Naïve Bayes, Logistic Regression, and Support vector machines with accuracy of \_\_, \_\_, \_\_ respectively, It is derived that Naïve Bayes has a better accuracy in predicting the likelihood of having a heart disease from new data presented to it. It will be the best model in implementing any system that uses machine learning in heart disease detection.

**5.5 REFERENCE**

Tom M. Mitchell (2006). The discipline of machine learning.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques.

Kenneth Dickstein et al., Warning Signs, Retrieved February 22, 2017, from <http://http://www.heartfailurematters.org/en_GB/Warning-signs/>

Margaret Rouse (2016) What is machine learning – Definition from WhatIs.com, Retrieved January 31, 2017, from <http://whatis.techtarget.com/definition/machine-learning>

Machine learning - Wikipedia (2017), Retrieved February 25, 2017, from <https://en.wikipedia.org/wiki/Machine_learning>

Timeline of Machine learning (2017), Retrieved February 25, 2017, from <https://en.wikipedia.org/wiki/Timeline_of_machine_learning>

Arthur Samuel - Wikipedia (2017), Retrieved February 25, 2017, from <https://en.wikipedia.org/wiki/Arthur_Samuel>

Frank Rosenblatt (1958). The Perceptron: A probabilistic model for information storage and organization in the brain.

Marvin Minsky and Seymour Papert, 1972 (2nd edition with corrections, first edition 1969) *Perceptrons: An Introduction to Computational Geometry*, The MIT Press, Cambridge MA, [ISBN 0-262-63022-2](https://en.wikipedia.org/wiki/Special:BookSources/0262630222).

Python data analysis library (2017), Retrieved June 28, 2017, from <http://pandas.pydata.org>

David E Rumelhart, Geoffrey E. Hinton, Ronald J. Williams, (9 October 1986). Retrieved June 5, 2016, from <http://elderlab.yorku.ca/~elder/teaching/cosc6390psyc6225/readings/hinton%201986.pdf>

Pandas Data Analysis Library (2017), Retrieved March 12, 2017, from <http://pandas.pydata.org/>

scikit-learn: machine learning in Python (2017), Retrieved March 12, 2017, from <http://scikit-learn.org/stable/>

Jason Brownlee (2013a) How to prepare data for machine learning, Retrieved December 16, 2017, from https://machinelearningmastery.com/how-to-prepare-data-for-machine-learning/

Jason Brownlee (2013b) How to evaluate machine learning algorithms, Retrieved December 23, 2017, from <https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/>

Jason Brownlee (2016) Visualize Machine Learning Data in Python With Pandas, Retrieved January 20, 2018, from <https://machinelearningmastery.com/visualize-machine-learning-data-python-pandas/>

Software maintenance – Wikipedia (2017) Retrieved December 24, 2017, from https://en.wikipedia.org/wiki/Software\_maintenance

David W, Aha (2016) UCI Machine Learning Repository: Heart Disease Data Set, Retrieved July 26, 2016, from http://archive.ics.uci.edu/ml/datasets/Heart+Disease